



NESNEYE YÖNELİK PROGRAMLAMA

Sipariş Otomasyon Sistemi

Grup Üyeleri

Engincan VESKE 172803020

Ömer Emre ELMALİ 162805010

İhsan GÜÇ 172803026

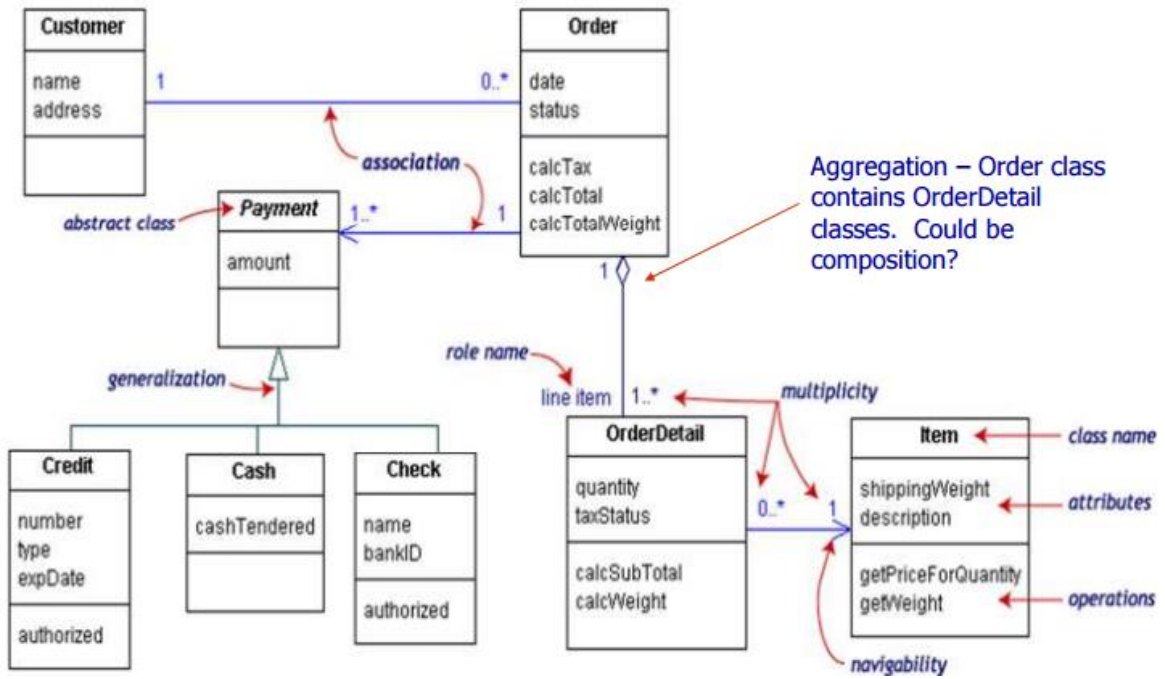
İçindekiler

- I. Projenin Amacı
- II. Giriş
- III. Kodlar ve Açıklamalar
- IV. Yöntem
- V. Sonuç
- VI. Kaynakça

Projenin Amacı

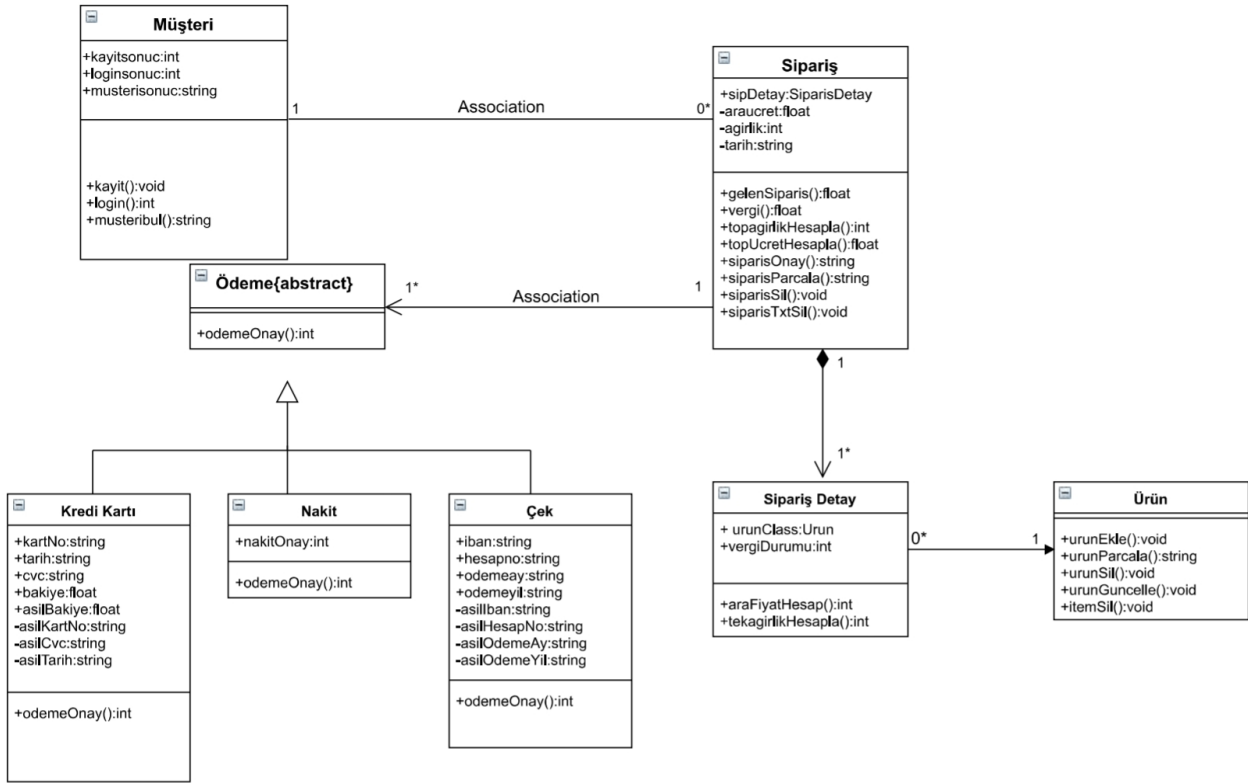
Basit, sade ve temiz kodlanmış bir sipariş otomasyon sistemi geliştirmek.

Giriş



Verilmiş olan UML Sınıf Diyagramı baz alınarak projenin gerekli implementasyonları yapılmış ve bu yapılan düzeltmeler düzenlenmiş olan UML Sınıf Diyagramında belirtilmiştir.

Yapılan düzenlemeler aşağıdaki UML diyagramında belirtilmiştir:



- Bize verilen UML diyagramındaki Sipariş ve Sipariş Detay sınıfları arasındaki aggregation olan ilişkiyi composition olarak düzenledik.
- Programın işlevselliğini arttırmak için planlayıp, uyguladığımız field(alan) ve methodları(metotları) programımıza ekledik.
- Ödeme sınıfından yararlanarak ödeme sınıflarında sınıfı override ederek(ezerek) her child (alt) sınıfta farklı bir ödeme işlemi tanımlamış olduk.
- Ayrıca ödeme işlemlerinde toplam tutarı en az 2 – en fazla 3 parçaya bölerek müşteriye daha çok ödeme imkanı sağlamış olduk.
- Bu yaptığımız otomasyon sisteminin amacı; müşteriye daha hızlı ve kolay(anlaşılır) bir biçimde, işlemini tamamlamasını sağlamaktır.

Kodlar ve Açıklamalar

```
item.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
Dar Pantolon+LC Waikiki+19.99+200+Bayan+1
Pantolon+Defacto+25+150+Bay+1
Gömlek+Defacto+28.88+168+Bayan+1
|
```

Ürün bilgileri item.txt dosyasının içinde başlıklara göre sıralanmış şekilde saklanmaktadır. Her satır bir ürünü belirtmektedir ve ürünlerin sütunları + ile ayrılmıştır.

```
user.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
|a+admin+admin+adres+05551231212+admin@admin.com+1
Ömer Emre Elmalı+oeelmali+oeelmali123+KYK Yurt+05511628184+omeremreelma@gmail.com+0
Engincan Veske+engincanv+oeelmali123+KYK Yurt+05558984096+enginveske@gmail.com+0
İhsan Güç+ihsan+ihsan+KYK Yurt+05534982127+beyih.ig@gmail.com+0
```

Aynı şekilde kullanıcı bilgileri de bir text dosyasının içinde saklanmaktadır. Yine kullanıcı bilgi sütunları + karakteri ile birbirinden ayrılmakta ve bu karakter ile program içerisinde bilgileri ayırmamızda bize yardımcı oluyor.

Ürünleri parçaladığımız kod bloğunu inceleyecek olursak:

```
public string[] urunparcala(string gelen)
{
    string[] dondururun = new string[6]; int artisay = 0;
    for (int j = 0; j < gelen.Length; j++)
    {
        if (gelen[j] == '+')
        {
            artisay++;
        }
        else if (artisay == 0)
        {
            dondururun[0] = dondururun[0] + gelen[j].ToString(); ;
        }
        else if (artisay == 1)
        {
            dondururun[1] = dondururun[1] + gelen[j];
        }
        else if (artisay == 2)
```

```

        {
            dondururun[2] = dondururun[2] + gelen[j];
        }
        else if (artisay == 3)
        {
            dondururun[3] = dondururun[3] + gelen[j];
        }
        else if (artisay == 4)
        {
            dondururun[4] = dondururun[4] + gelen[j];
        }
        else if (artisay == 5)
        {
            dondururun[5] = dondururun[5] + gelen[j];
        }
    }

    return dondururun;
}

```

Yukarıdaki urunparcala() fonksiyonuna text dosyasındaki her satırı teker teker yollayarak ürün bilgilerinin parçalanarak string dizisine atılması sağlanır. Aynı şekilde siparisparcala() ve login() de aynı şekilde çalışmaktadır.

Müşteriyi sisteme kayıt etmek için ise user.txt dosyasına eklemek için aşağıdaki kod satırları kullanılmaktadır.

```

public void kayit(string ad,string kulad, string sifre, string tekrarsifre,string adres,string tel,
string email,string kt)
{
    int sonuc = 1;
    if (ad == "" || ad == null || kulad == "" || kulad == null || sifre == "" || sifre ==
null || tekrarsifre == null || tekrarsifre == "" || adres== null || adres == "" || tel == "" || tel
== null || email== "" || email == null)
    {
        sonuc = 0;
        MessageBox.Show("Lütfen boş kutucukları doldurunuz.");
    }
    if (sifre != tekrarsifre)
    {
        sonuc = 0;
        MessageBox.Show("Şifreler uyuşmuyor lütfen kontrol ediniz.");
    }
    if (sonuc==1)
    {
        string kayit = ad + "+" + kulad + "+" + sifre + "+" + adres + "+" + tel + "+" +
email + "+" + kt + Environment.NewLine;
        File.AppendAllText("user.txt", kayit);
    }
}

```

musteri.cs sınıfının içerisindeki kayit() fonksiyonunun içine tüm gerekli bilgiler gönderildiği takdirde kullanıcı istenilen formatta text dosyasının içerisine başarılı olarak kayıt edilir.

```

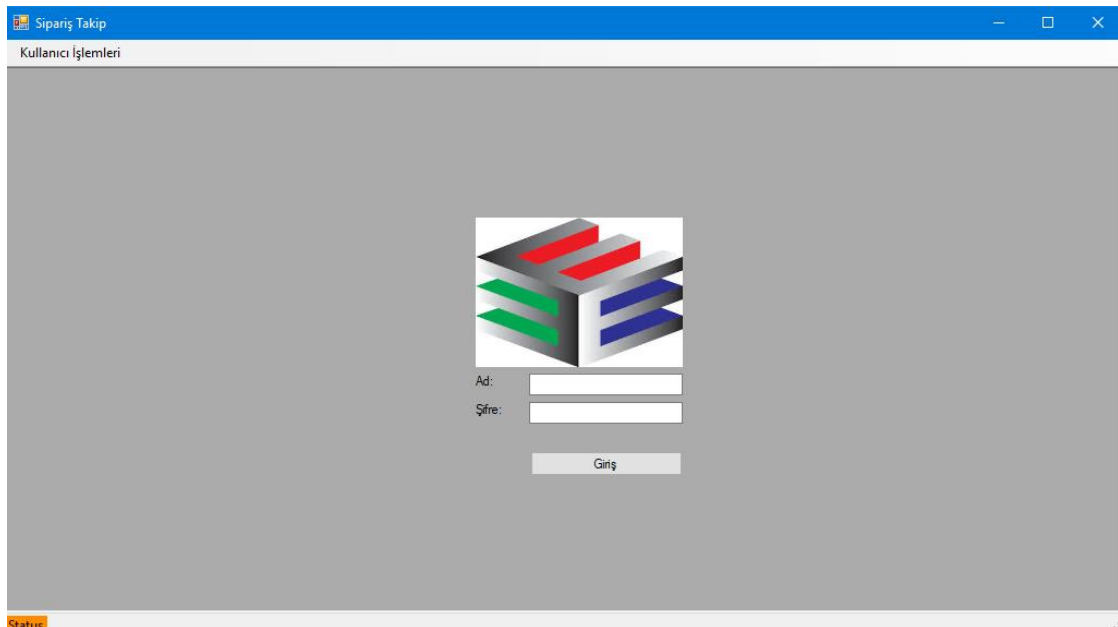
public void siparissil(string urun)
{
    string line = null;
    using (StreamReader reader = new StreamReader("siparis.txt"))
    {
        using (StreamWriter writer = new StreamWriter("siparis1.txt"))
        {
            while ((line = reader.ReadLine()) != null)
            {
                if (String.Compare(line, urun) == 0)
                    continue;

                writer.WriteLine(line);
            }
            writer.Close();
        }
        reader.Close();
    }
    siparistxtsil();
}
void siparistxtsil()
{
    if (File.Exists("siparis.txt"))
    {
        File.Delete("siparis.txt");
    }
    FileInfo info = new FileInfo("siparis1.txt");
    info.MoveTo("siparis.txt");
}

```

Yukarıdaki iki fonksiyon ise sipariş silmek için gerekli işlemleri gerçekleştirir. Silinecek olan satır önceden form içerisinde birleştirilerek siparissil() fonksiyonunun içerisine gönderilir ve text dosyasında bu satıra karşılık gelen kısım atlanarak yeni bir text1 dosyasına yazılır. En sonra text dosyası silinir ve text1 dosyasının adı text olarak değiştirilir böylelikle istenilen sipariş silinir. Bununla aynı şekilde ürün silme işlemi de bu fonksiyonlarla aynı işlemleri gerçekleştirir.

Programın Formları (Sayfaları)



Sipariş Takip

Kullanıcı İşlemleri

Kayıt Sayfası

Ad Soyad

E-mail

Telefon

Kullanıcı Adı

Şifre

Tekrar Şifre

Adres

Kayıt Ol

Status

Sipariş Takip

Kullanıcı İşlemleri

yonetici

Ürün Bilgileri Ekle Güncelle Sipariş

Ürünler	Marka	Fiyat	Stok	Cinsiyet	Ağırlık
Dar Pantolon	LC Wai...	19.99	200	Bayan	1
Pantolon	Defacto	25	150	Bay	1
Gömlek	Defacto	28.88	168	Bayan	1

Başlık

Beden

Renk

Marka

Fiyat

Adet

Güncelle

Ekle

Sil

Status

Sipariş Takip

Kullanıcı İşlemleri

yonetici

Ürün Bilgileri Ekle Güncelle Sipariş

Ürün Adı

Marka

Fiyat

Stok

Ağırlık

Cinsiyet

Güncelle

Status

Sipariş Takip

Kullanıcı İşlemleri

yonetici

Ürün BilgileriEkleGüncelleSipariş

Tarih	Ürünler	Alıcı	Alıcı Adres	Odenen Tutar
-------	---------	-------	-------------	--------------

Sipariş Sil

Status


Sipariş Takip

Kullanıcı İşlemleri

musteri

Ürün BilgileriSepetÖdeme İşlemleri

Ürünler	Marka	Fiyat	Stok	Cinsiyet	Ağırlık
Dar Pantolon	LC Wai...	19.99	200	Bayan	1
Pantolon	Defacto	25	150	Bay	1
Gömlek	Defacto	28.88	168	Bayan	1



Bağlık

Beden

Renk

Marka

Cinsiyet

Fiyat

Adet

1

Sepete Ekle

Vergi indirim kodunuz var ise giriniz

Kodu Onayla

Status

Sipariş Takip

Kullanıcı İşlemleri

musteri

Ürün BilgileriSepetÖdeme İşlemleri

Ürün	Marka	Fiyat	Stok	Cinsiyet	Ağırlık	Adet	Beden	Renk
Gömlek	Defacto	28.88	168	Bayan	1	1	S	Kahver...

Ara Toplam: 28.88 TL

Ürün Sil

İşlemi Onayla

36.0784

TL Olan fiyatınıza Kargo Bedeli + %18 KDV Dahildir. Eğer vergi indirim kodu girdiyseiz indirim uygulanmıştır.

Status

Sipariş Takip

Kullanıcı İşlemleri

musteri

Ürün BilgileriSepetÖdeme İşlemleri

36.0784

Kredi KartıNakitÇekParçalı Ödeme

☐ Kredi Kartı☐ Nakit☐ Çek

☐ Kapıda ödeme (+%5) olduğunu kabul ediyorum

Kart NoKart TarihKart CVCİban NoHesap NoÇek Tarih

Ödemeyi Tamamla

Kredi kartı bilgileri : Kart No : 12345 | Tarih : 07/22 CVC : 150 - Çek Bilgileri : Hesap ve İban : 12345 Tarih : 07/22

Status

Sipariş Takip

Kullanıcı İşlemleri

musteri

Ürün BilgileriSepetÖdeme İşlemleri

36.0784

Kredi KartıNakitÇekParçalı Ödeme

İban NoHesap NoÇek Tarih

Ödemeyi Tamamla

Kredi kartı bilgileri : Kart No : 12345 | Tarih : 07/22 CVC : 150 - Çek Bilgileri : Hesap ve İban : 12345 Tarih : 07/22

Status

Sipariş Takip

Kullanıcı İşlemleri

musteri

Ürün Bilgileri Sepet Ödeme İşlemleri

37.88232

Kredi Kartı Nakit Çek Parçalı Ödeme

☐ Kapıda ödeme yapmayı ve Kapıda ödeme için %5 ek ücret ödemeyi kabul ediyorum

Ödemeyi Tamamla

Kredi kartı bilgileri : Kart No : 12345 | Tarih : 07/22 CVC : 150 - Çek Bilgileri : Hesap ve İban : 12345 Tarih : 07/22

Status

Sipariş Takip

Kullanıcı İşlemleri

musteri

Ürün Bilgileri Sepet Ödeme İşlemleri

36.0784

Kredi Kartı Nakit Çek Parçalı Ödeme

Kart No

Kart Tarih

Kart CVC

Ödemeyi Tamamla

Kredi kartı bilgileri : Kart No : 12345 | Tarih : 07/22 CVC : 150 - Çek Bilgileri : Hesap ve İban : 12345 Tarih : 07/22

Status

Yöntem

C# programlama dili kullanılarak Nesneye Yönelik Programlama(OOP) ve Katmanlı Mimari yazılım prensibi uygulanarak proje hazırlanılmıştır.

Sonuç

- Geliştirdiğimiz bu proje sayesinde projede kullandığımız C# dili ve UML diyagramındaki Association ilişki türlerinden olan Aggregation ve Composition ilişkilerini uygulayabildik.
- Önceden bize verilen Aggregation ilişkisini Composition olarak değiştirdik ve daha etkili olduğunu saptadık. Bunun nedeni bir sipariş olmadan sipariş detayının oluşamayacağını düşünmemizdir.
- Verilen UML diyagramını daha kısa ve açıklayıcı yaparak kullanan müşterilerin anlamasına olanak sağladık .

KAYNAKÇA

- Doç.Dr.Fahri VATANSEVER-Algoritma Geliştirme Ve Programlama(sayfa 399)-Dosya işlemleri için kullanıldı.
- Microsoft Documentation (ListView Özellikleri) kullanıldı.
- Guru99.com (MDI Class özellikleri) kullanıldı.